Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation

Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation

Roman Grundkiewicz
University of Edinburgh
10 Crichton St, Edinburgh EH8 9AB, Scotland
rgrundki@inf.ed.ac.uk

Marcin Junczys-Dowmunt
Microsoft
Redmond, WA 98052, USA
marcinjd@microsoft.com

Abstract

We combine two of the most popular approaches to automated Grammatical Error Correction (GEC): GEC based on Statistical Machine Translation (SMT) and GEC based on Neural Machine Translation (NMT). The hybrid system achieves new state-of-the-art results on the CoNLL-2014 and JFLEG benchmarks. This GEC system preserves the accuracy of SMT output and, at the same time, generates more fluent sentences as it typical for NMT. Our analysis shows that the created systems are closer to reaching human-level performance than any other GEC system reported so far.

1 Introduction

Currently, the most effective GEC systems are based on phrase-based statistical machine translation (Rozovskaya and Roth, 2016; Junczys-Dowmunt and Grundkiewicz, 2016; Chollampatt and Ng, 2017). Systems that rely on neural machine translation (Yuan and Briscoe, 2016; Xie et al., 2016; Schmaltz et al., 2017; Ji et al., 2017) are not yet able to achieve as high performance as SMT systems according to automatic evaluation metrics (see Table 1 for comparison on the CoNLL-2014 test set). However, it has been shown that the neural approach can produce more fluent output, which might be desirable by human evaluators (Napoles et al., 2017). In this work, we combine both MT flavors within a hybrid GEC system. Such a GEC system preserves the accuracy of SMT output and at the same time generates more fluent sentences achieving new state-of-the-art results on two different benchmarks: the annotation-based CoNLL-2014 and the fluency-based JFLEG benchmark. Moreover, comparison with human gold standards shows that the created systems are closer to reaching human-level performance than any other GEC system described in the literature so far.

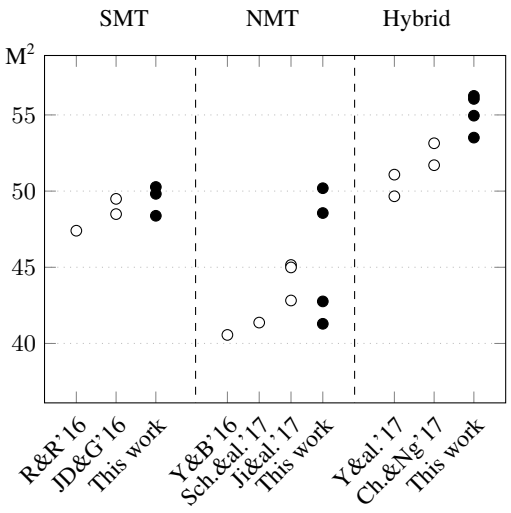

Figure 1: Comparison of SMT, NMT and hybrid GEC systems on the CoNLL-2014 test set (M^2).

Using consistent training data and preprocessing (§ 2), we first create strong SMT (§ 3) and NMT (§ 4) baseline systems. Then, we experiment with system combinations through pipelining and reranking (§ 5). Finally, we compare the performance with human annotations and identify issues with current state-of-the-art systems (§ 6).

2 Data and preprocessing

Our main training data is NUCLE (Dahlmeier et al., 2013). English sentences from the publicly available Lang-8 Corpora (Mizumoto et al., 2012) serve as additional training data.

We use official test sets from two CoNLL shared tasks from 2013 and 2014 (Ng et al., 2013, 2014) as development and test data, and evaluate using M^2 (Dahlmeier and Ng, 2012). We also report results on JFLEG (Napoles et al., 2017) with the

Corpus	Sentences	Tokens
NUCLE	57,151	1,162K
Lang-8 NAIST	1,943,901	25,026K
CoNLL-2013 (dev)	1,381	29K
CoNLL-2014 (test)	1,312	30K
JFLEG Dev	754	14K
JFLEG Test	747	13K

Table 1: Statistics for training and testing data sets.

System	CoNLL P	CoNLL R	CoNLL M^2	JFLEG GLEU
SMT Dense	56.91	30.25	48.38	54.68
+ Sparse	60.28	29.40	**49.82**	55.25
+ Char. ops	60.27	30.21	**50.27**	**55.79**

Table 2: Results for SMT baseline systems on the CoNLL-2014 (M^2) and JFLEG Test (GLEU) sets.

GLEU metric (Napoles et al., 2015). The data set is provided with a development and test set split. All data sets are listed in Table 1.

We preprocess Lang-8 with the NLTK tokenizer (Bird and Loper, 2004) and preserve the original tokenization in NUCLE and JFLEG. Sentences are truecased with scripts from Moses (Koehn et al., 2007). For dealing with out-of-vocabulary words, we split tokens into 50k subword units using Byte Pair Encoding (BPE) by Sennrich et al. (2016b). BPE codes are extracted only from correct sentences from Lang-8 and NUCLE.

3 SMT systems

For our SMT-based systems, we follow recipes proposed by Junczys-Dowmunt and Grundkiewicz (2016), and use a phrase-based SMT system with a log-linear combination of task-specific features. We use word-level Levenshtein distance and edit operation counts as dense features (Dense), and correction patterns on words with one word left/right context on Word Classes (WC) as sparse features (Sparse). We also experiment with additional character-level dense features (Char. ops). All systems use a 5-gram Language Model (LM) and OSM (Durrani et al., 2011) both estimated from the target side of the training data, and a 5-gram LM and 9-gram WCLM trained on Common Crawl data (Buck et al., 2014).

Experiment settings Translation models are trained with Moses (Koehn et al., 2007), word-alignment models are produced with MGIZA++ (Gao and Vogel, 2008), and no reordering models are used. Language models are built using KenLM (Heafield, 2011), while word classes are trained with word2vec[1].

We tune the systems separately for M^2 and GLEU metrics. MERT (Och, 2003) is used for tuning dense features and Batch Mira (Cherry and Foster, 2012) for sparse features. For M^2 tunning we follow the 4-fold cross-validation on NUCLE with adapted error rate recommended by Junczys-Dowmunt and Grundkiewicz (2016). Models evaluated on GLEU are optimized on JFLEG Dev using the GLEU scorer, which we added to Moses. We report results for models using feature weights averaged over 4 tuning runs.

Results Other things being equal, using the original tokenization, applying subword units, and extending edit-based features result in a similar system to Junczys-Dowmunt and Grundkiewicz (2016): 49.82 vs 49.49 M^2 (Table 2).

The phrase-based SMT systems do not deal well with orthographic errors (Napoles et al., 2017) — if a source word has not been seen in the training corpus, it is likely copied as a target word. Subword units can help to solve this problem partially. Adding features based on character-level edit counts increases the results on both test sets.

A result of 55.79 GLEU on JFLEG Test is already 2 points better than the GLEU-tuned NMT system of Sakaguchi et al. (2017) and only 1 point worse than the best reported result by Chollampatt and Ng (2017) with their M^2-tuned SMT system, even though no additional spelling correction has been used at this point. We experiment with specialized spell-checking methods in later sections.

4 NMT systems

The model architecture we choose for our NMT-based systems is an attentional encoder-decoder model with a bidirectional single-layer encoder and decoder, both using GRUs as their RNN variants (Sennrich et al., 2017). A similar architecture has been already tested for the GEC task by Sakaguchi et al. (2017), but we use different hyperparameters.

To improve the performance of our NMT models, similarly to Xie et al. (2016) and Ji et al. (2017), we combine them with an additional large-scale language model. In contrast to previous studies, which use an n-gram probabilistic LM, we build a 2-layer Recurrent Neural Network Language Model (RNN

[1] https://github.com/dav/word2vec

System	CoNLL P	CoNLL R	CoNLL M^2	JFLEG GLEU
NMT	66.61	17.58	42.76	50.08
NMT + RNN-LM	61.05	**26.71**	48.56	56.04
NMT×4	**71.10**	15.42	41.29	50.30
NMT×4 + RNN-LM	60.27	30.08	**50.19**	**56.74**

Table 3: Results for NMT systems on the CoNLL-2014 (M^2) and JFLEG Test (GLEU) sets.

LM) with GRU cells which we train again on English Common Crawl data (Buck et al., 2014).

Experimental settings We train with the Marian toolkit (Junczys-Dowmunt et al., 2018) on the same data we used for the SMT baselines, i.e. NUCLE and Lang-8. The RNN hidden state size is set to 1024, embedding size to 512. Source and target vocabularies as well as subword units are the same.

Optimization is performed with Adam (Kingma and Ba, 2014) and the mini-batch size fitted into 4GB of GPU memory. We regularize the model with scaling dropout (Gal and Ghahramani, 2016) with a dropout probability of 0.2 on all RNN inputs and states. Apart from that we dropout entire source and target words with probabilities of 0.2 and 0.1 respectively. We use early stopping with a patience of 10 based on the cross-entropy cost on the CoNLL-2013 test set. Models are validated and saved every 10,000 mini-batches. As final models we choose the one with the best performance on the development set among the last ten model check-points based on the M^2 or GLEU metrics.

Size of RNN hidden state and embeddings, target vocabulary, and optimization options for the RNN LM are identical to those used for our NMT models. Decoding is done by beam search with a beam size of 12. We normalize scores for each hypothesis by sentence length.

Results A single NMT model achieves lower performance than the SMT baselines (Table 3). However, the M^2 score of 42.76 for CoNLL-2014 is already higher than the best published result of 41.53 M^2 for a strictly neural GEC system of Ji et al. (2017) that has not been enhanced by an additional language model.

Our RNN LM is integrated with NMT models through ensemble decoding (Sennrich et al., 2016a). Similarly to Ji et al. (2017), we choose the weight of the language model using grid search on the development set[2]. This strongly improves recall,

[2]Used weights are 0.2 and 0.25 for M^2 and GLEU evaluation, respectively.

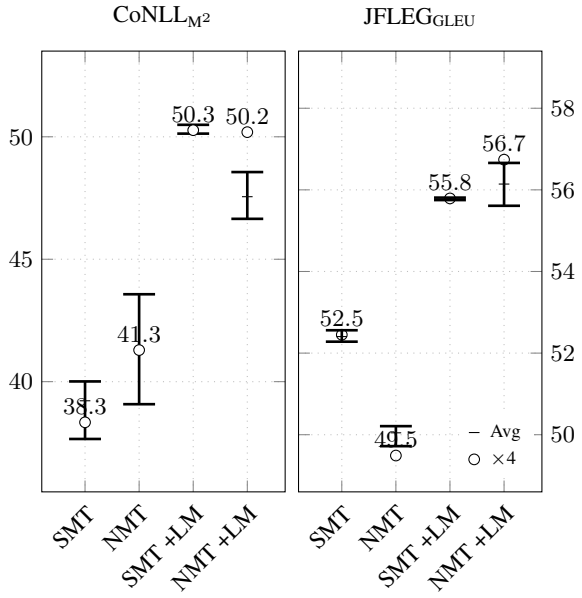

Figure 2: Contribution of a language model (LM) for SMT and NMT GEC systems.

and thus boosts the results significantly on both test sets (+5.8 M^2 and +5.96 GLEU).

An ensemble of four independently trained models[3] (NMT×4), on the other hand, increases precision at the expense of recall, which may even lead to a performance drop. Adding the RNN LM to that ensemble balances this negative effect, resulting in 50.19 M^2. These are by far the highest results reported on both benchmarks for pure neural GEC systems.

Comparison to SMT systems With model ensembling, the neural systems achieve performance similar to SMT baselines (Figure 2). A stripped-down SMT system without CCLM, quite surprisingly gives better results on JFLEG than the NMT system, and the opposite is true for CoNLL-2014. The reason for the lower performance on JFLEG might be a large amount of spelling errors, which are more efficiently corrected by the SMT system using subword units.

If both systems are enhanced by a large-scale language model, the neural system outperforms the SMT system on JFLEG and it is competitive with SMT systems on CoNLL-2014. However, it is not known if the results would preserve if the NMT model is combined with a probabilistic n-gram LM instead as it has been proposed in the previous works (Xie et al., 2016; Ji et al., 2017).

[3]Each model is weighted equally during decoding.

System	CoNLL P	CoNLL R	CoNLL M^2	JFLEG GLEU
Best SMT	60.27	30.21	50.27	55.79
→ Pip. NMT	60.25	34.80	52.56	57.21
→ Pip. NMT+LM	58.87	**39.23**	53.51	**58.83**
+ Res. RNN-LM	70.97	24.86	51.77	56.97
+ Res. NMT	70.40	26.69	53.03	57.21
+ Res. NMT+LM	71.40	**28.60**	**54.95**	57.53
→ Pip. NMT+LM	65.73	33.36	**55.05**	**58.83**
+ Spell SMT	70.80	30.57	56.05	60.09
→ Pip. NMT+LM	66.77	34.49	**56.25**	**61.50**

Table 4: Results for hybrid SMT-NMT systems on the CoNLL-2014 (M^2) and JFLEG Test (GLEU) sets.

5 Hybrid SMT-NMT systems

We experiment with pipelining and rescoring methods in order to combine our best SMT and NMT GEC systems[4].

SMT-NMT pipelines The output corrected by an SMT system is passed as an input to the NMT ensemble with or without RNN LM[5]. In this case the NMT system serves as an automatic post-editing system. Pipelining improves the results on both test sets by increasing recall (Table 4). As the performance of the NMT system without a RNN LM is much lower than the performance of the SMT system alone, this implies that both approaches produce complementary corrections.

Rescoring with NMT Rescoring of an n-best list obtained from one system by another is a commonly used technique in GEC, which allows to combine multiple different systems or even different approaches (Hoang et al., 2016; Yannakoudakis et al., 2017; Chollampatt and Ng, 2017; Ji et al., 2017). In our experiments, we generate a 1000 n-best list with the SMT system and add separate scores from each neural component. Scores of NMT models and the RNN LM are added in the form of probabilities in negative log space. The re-scored weights are obtained from a single run of the Batch Mira algorithm (Cherry and Foster, 2012) on the development set.

As opposed to pipelining, rescoring improves precision at the expense of recall and is more effective for the CoNLL data resulting in up to 54.95 M^2. On JFLEG, rescoring only with the RNN LM produces similar results as rescoring with the NMT ensemble. However, the best result for rescoring is lower than for pipelining on that test set. It seems the SMT system is not able to produce as diversified corrections in an n-best list as those generated by the NMT ensemble.

[4]The best system combinations are chosen again based on the development sets, i.e. CoNLL-2013 and JFLEG Dev. We omit these results as they are highly overestimated.

[5]We did not observed any improvements if the order of the systems is reversed.

System	CoNLL-10 P	CoNLL-10 R	CoNLL-10 M^2	JFLEG GLEU
Human Avg.	73.17	**68.75**	72.15	62.38
Ch&Ng'17	79.46	43.73	68.29	56.78
Ratio (%)	1.08	0.64	94.66	91.02
This work	**83.15**	46.97	72.04	61.50
Ratio (%)	1.14	0.68	**99.85**	**98.59**

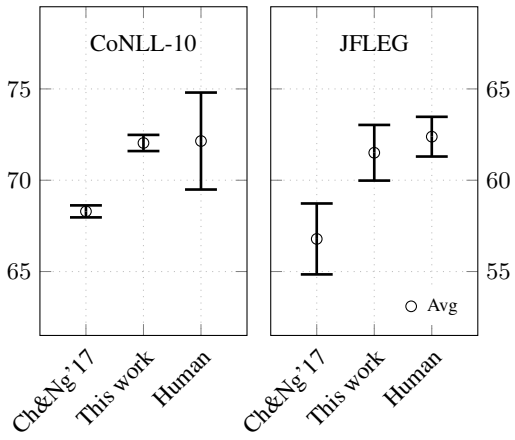

Figure 3: Comparison with human annotators. The figure presents average M^2 and GLEU scores with standard deviations.

Spelling correction and final results Pipelining the NMT-rescored SMT system and the NMT system leads to further improvement. We believe this can be explained by different contributions to precision and recall trade-offs for the two methods, similar to effects observed for the combination of the NMT ensemble and our RNN LM.

On top of our final hybrid system we add a spell-checking component, which is run before pipelining. We use a character-level SMT system following Chollampatt and Ng (2017) which they deploy for unknown words in their word-based SMT system. As our BPE-based SMT does not really suffer from unknown words, we run the spell-checking component on words that would have been segmented by the BPE algorithm. This last system achieves the best results reported in this paper: 56.25 M^2 on CoNLL-2014 and 61.50 GLEU on JFLEG Test.

System	Example
Source	*but now every thing is change , the life becom more dificullty .*
Best SMT	*But now **everything is changed** , the life becom more dificullty .*
Best NMT	*But now **everything is changing** , the life **becomes** more **difficult** .*
Pipeline	*But now **everything is changed** , the life **becomes** more **difficult** .*
Rescoring	*But now **everything has changed** , the life becom more dificullty .*
+ Pipeline	*But now **everything has changed** , the life **becomes** more **difficult** .*
Reference 1	*Now everything has changed , and life becomes more difficult .*
Reference 2	*Everything has changed now and life has become more difficult .*
Reference 3	*But now that everything changes , life becomes more difficult .*
Reference 4	*But now that everything is changing , life becomes more difficult .*

Table 5: System outputs for the example source sentence from the JFLEG Test set.

6 Analysis and future work

For both benchmarks our systems are close to automatic evaluation results that have been claimed to correspond to human-level performance on the CoNLL-2014 test set and on JFLEG Test.

Example outputs Table 5 shows system outputs for an example source sentence from the JFLEG Test corpus that illustrate the complementarity of the statistical and neural approaches. The SMT and NMT systems produce different corrections. Rescoring is able to generate a unique correction (*is change→has changed*), but it fails in generating some corrections from the neural system, e.g. misspellings (*becom* and *dificullty*). Pipelining, on the other hand, may not improve a local correction made by the SMT system (*is changed*). The combination of the two methods produces output, which is most similar to the references.

Comparison with human annotations Bryant and Ng (2015) created an extension of the CoNLL-2014 test set with 10 annotators in total, JFLEG already incorporates corrections from 4 annotators. Human-level results for M^2 and GLEU were calculated by averaging the scores for each annotator with regard to the remaining 9 (CoNLL) or 3 (JFLEG) annotators, respectively.

Figure 3 contains human level scores, our results, and previously best reported results by Chollampatt and Ng (2017). Our best system reaches nearly 100% of the average human score according to M^2 and nearly 99% for GLEU being much closer to that bound than previous works[6].

Further inspection reveals, however, that the precision/recall trade-off for the automatic system indicates lower coverage compared to human corrections — lower recall is compensated with high precision[7]. Automatic systems might, for example, miss some obvious error corrections and therefore easily be distinguishable from human references. Future work would require a human evaluation effort to draw more conclusions.

Acknowledgments

This work was partially funded by Facebook. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Facebook.

References

Steven Bird and Edward Loper. 2004. NLTK: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, page 31.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 697–707. http://www.aclweb.org/anthology/P15-1068.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the Common Crawl. In *Proceedings of the Language Resources and Evaluation Conference*. Reykjavík, Iceland, pages 3579–3584.

[6] During the camera-ready preparation, Chollampatt and Ng (2018) have published a GEC system based on a multilayer convolutional encoder-decoder neural network with a character-based spell-checking module improving the previous best result to 54.79 M^2 on CoNLL-2014 and 57.47 GLEU on JFLEG Test.

[7] A similar imbalance between precision and recall is visible on JFLEG when the M^2 metric is used.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Stroudsburg, USA, pages 427–436.

Shamil Chollampatt and Hwee Tou Ng. 2017. Connecting the dots: towards human-level grammatical error correction. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Copenhagen, Denmark, pages 327–333. http://www.aclweb.org/anthology/W17-5037.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 568–572.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The NUS corpus of learner english. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. pages 22–31.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 1045–1054. http://www.aclweb.org/anthology/P11-1105.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*. pages 1019–1027.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*. ACL, pages 49–57.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Stroudsburg, USA, WMT '11, pages 187–197.

Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an SMT approach to grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI/AAAI Press, pages 2803–2809.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 753–762.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1546–1556. https://aclweb.org/anthology/D16-1161.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. *arXiv preprint arXiv:1804.00344* https://arxiv.org/abs/1804.00344.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* .

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics*. The Association for Computer Linguistics.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yu Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012*. pages 863–872.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 588–593. http://www.aclweb.org/anthology/P15-2097.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 2017 Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain. https://arxiv.org/abs/1702.04066.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and

Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Baltimore, Maryland, pages 1–14. http://www.aclweb.org/anthology/W14-1701.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1–12. http://www.aclweb.org/anthology/W13-3601.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, USA, ACL '03, pages 160–167.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2205–2215. http://www.aclweb.org/anthology/P16-1208.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, pages 366–372. http://www.aclweb.org/anthology/I17-2062.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting sequence models for sentence correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2807–2813. https://www.aclweb.org/anthology/D17-1298.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain, pages 65–68. http://aclweb.org/anthology/E17-3017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 371–376. http://www.aclweb.org/anthology/W/W16/W16-2323.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. http://www.aclweb.org/anthology/P16-1162.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727* .

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural sequence-labelling models for grammatical error correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2785–2796. https://www.aclweb.org/anthology/D17-1296.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 380–386.